Video Games for the Brain

Leveraging Unreal Engine to increase cognitive task engagement

Ian L. Jackson | Duke University | Electrical & Computer Engineering (MS)

# Introduction

The Cogan Lab investigates the neural bases of speech perception and production. Much of the data used for this research is collected from sessions with patients who have intracranial stereoelectroencephalography (sEEG) electrodes implanted for seizure localization as part of presurgical epilepsy monitoring. These data collection sessions involve tasks that assess speech repetition, lexical decision (discerning words from non-words), verbal working memory, and other language-related cognitive functions. Because of the amount of data required for reliable analysis of neural correlates, these tasks are comprised of a large number of trials. The lab’s Sternberg Neighborhood task, for example, consists of 5 blocks, each having 32 trials, totaling 160 trials over the roughly 45-minute session. During each trial, a participant is tasked with the same goal of determining whether a spoken word or non-word was part of a list defined at the beginning of the block. Other tasks are similar in duration (typically between 30 to 60 minutes) and equally rote in nature. The lengthiness and repetitiveness of these tasks result in a marked lack of engagement with the tasks over the duration of a session, especially in adolescent participants (7-14 years old), contributing to mental fatigue and decreased inattention.

Several studies investigating sustained attention[[1]](#footnote-2) using the Sustained Attention to Response Task (SART)[[2]](#footnote-3) show that lapses of engagement (measured by self-reported mind wandering and the amount of time between responses) with the task correlates to lower performance [1-3], and is especially exacerbated by task repetitiveness. [4-5] Lapses of attention have also been tied to a decrease in working memory, reading comprehension, and other cognitive factors. [1] Especially relevant to the Cogan Lab, sustained attention is also evidenced to play a modulatory role in language production [6], location of neural activations in response to speech [7], and word recall [8]. Thus, the notably attenuated engagement during in-unit sessions with patients is ultimately detrimental to both the study itself and the participant’s overall experience. Furthermore, because adolescents – those most susceptible to distraction – account for about 30-45% of all participants, their inclusion is important to the robustness of any findings from the studies. It is, then, in the best interest of future studies to increase task engagement.

As a solution to this problem, we seek to utilize video games with built-in language and other cognitive tasks to increase overall engagement. Gamification – the application of design principles from games (e.g. achieving “levels,” use of strategy, and in-game rules and rewards) to other contexts – is an increasingly popular approach in behavioral task design. Countless studies have turned to creating games as tasks for studying attention [9-11], memory [9, 12], learning [13-14], and more [15-16]. These design choices are supported by strong evidence that gamification increases motivation and engagement in participants compared to standard tasks. [9, 17] Moreover, recent technological advancements in high-resolution 3D rendering and virtual reality (VR) have redefined the bounds of gamification in the study of the brain. Due to the increased affordability of this technology, progressively more labs are utilizing their benefits. There is evidence that more dynamic, lifelike stimuli have been shown to increase engagement and enjoyment in tasks, further motivating the push for using rich 3D environments during brain recordings. [17] The hardware of the sEEG utilized in-unit, however, currently precludes the potential for using VR, as a standard VR headset would obstruct the wiring connecting the electrodes to the amplifiers. Nonetheless, the advantages of replacing the currently employed static task environments with more immersive ones are attainable with commercially available software.

At the benefit of being less computationally-demanding software, traditional frameworks for behavioral experiments like PsychoPy (Python), Presentation (Neurobehavioral Systems), and Psychtoolbox (MATLAB) are limited in their ability to generate graphics with the same resolution of high-quality, high-performance games that are popular today. Instead, to create the immersive 3D worlds that maximize user engagement, we turn to a more intensive software: Unreal Engine (Epic Games). Unreal Engine is a 3D game engine known for its highly detailed effects and animation capabilities, robust support for custom plugins, and general ubiquity in the field of game development. Because of these distinguishing factors, previous studies have utilized the platform to implement environments and tasks for behavioral experiments [18-19] as well as skills training. [20-21] No such studies, however, have attempted to synchronize in-game stimuli produced by Unreal Engine to live neural data recordings in order to study neural correlates. Using stimuli presented in an Unreal Engine-generated environment (e.g., text prompts for a word repetition task) and an entertaining design, we seek to leverage the active user engagement induced by an immersive 3D video game in order to better study the cognitive neuroscience of speech perception and production.

An essential step to implementing any stimulus presentation software for neuroscientific experiments is the accuracy and precision of the timing of those stimuli, specifically to synchronize the stream of real-time stimulus delivery with simultaneous recorded neural activity. A discrepancy of just a few milliseconds between the two can lead to severe inaccuracies in determining what stimuli gave rise to neural activity at a given time. In the context of studying language, these inaccuracies, if large enough, can mean the difference between entire phonemes or even words. Therefore, it is important to assess any timing errors of a stimulus delivery platform to then prevent and/or predict timing delays in practice. Due to the hardware and software limitations of stimulus delivery on a computer (e.g., frame rate, RAM), as well as the fast timescales on which the brain functions, the optimization of this stream synchronization has been a topic of study for decades. [22-24] While stimulus presentation packages like PsychoPy, Presentation, and PsychtoolBox have a long record of successful validation testing in the context of neuroscience research, few studies have investigated the precision and accuracy of visual stimulus delivery in Unreal Engine or its predecessors. [25-26] Furthermore, no studies to date have investigated the same characteristics for auditory stimulus delivery in Unreal Engine. Thus, this project has two goals: 1) to measure and optimize UE4 stimulus delivery timing for use in neuroscience experiments, and 2) to create video games that increase engagement and motivation in study participants.

# Materials & Methods

## Experimental Overview

The lab’s typical experimental setup for in-unit language tasks consists of a stimulus presentation laptop running a task on Psychtoolbox (MathWorks, Inc.) providing both auditory and visual stimuli. During in-unit sessions, a photodiode is placed over the top-right corner of the laptop screen, which is activated by an on-screen stimulus. Because the photodiode data is passed through the same recorder as the neural data, these flashes essentially mark the neural data in real-time. During post-session analysis these flashes can then be connected to event codes recorded during the session in order to correlate the neural response to a given stimulus. The testing sessions run for the project presented in this paper utilize a similar configuration to quantify stimulus delays (Figure 1), where the main modifications are the software presenting the stimuli, Unreal Engine version 4.27 (UE4), and the inclusion of a microphone to measure auditory stimuli. The photodiode and microphone streams are sent in parallel to a dummy neural signal, which are all then sent to a data acquisition device that saves the received data to a recording laptop. At the same time, the timestamp (relative to the start of the game) and event information of an in-game auditory or visual stimulus is recorded in a CSV file using a custom UE4 plugin written in C++. These parallel processes allow the same time mapping that is done in the in-unit processing pipeline, thus providing a window into the latency between the time of events recorded by the computer and time from the data acquisition system.

Graphical user interface, application, Teams

Description automatically generated

*Figure 1. Outline of the typical recording setup, where audio data is represented by green arrows, visual data by red arrows, and neural data by blue arrows. Passing the streams through a common data acquisition device allows for synchronization of auditory and visual stimuli with neural data, which is used to accurately correlate neural responses to the stimuli.*

All stimulus tests were run on a Lenovo Legion S7 (AMD Ryzen 7, 16GB DDR4 RAM, NVIDIA GeForce RTX 3060). This testing laptop had a refresh rate of 165Hz. Stimuli were recorded on a 1024-channel RHD recording controller (Intan Technologies) at a sampling rate of 20kHz. The recording controller was connected to a recording laptop, a Lenovo Thinkpad P50 (Intel Core i7-6700HQ, 32GB DDR4 RAM, NVIDIA Quadro 1000M) which saved all session data locally. Audio from the testing laptop was processed through an audio amplifier (FiiO Olympus 2 DAC), which was then connected to an external speaker (Anker Soundcore Motion Boom Plus). This configuration was used due to previous tests demonstrating decreased audio delivery latency with the inclusion of the speaker and amplifier. The audio from the speaker was then picked up by a microphone connected to an audio interface (Behringer U-Phoria UM22), which sent the detected sound to an analog-in port on the Intan device. Visual stimuli were detected by a mounted photodiode (ThorLabs, Inc.) attached to the top-right corner of the stimulus laptop screen with a custom 3D-printed attachment and connected to an analog-in port on the Intan.

## Visual Stimulus Testing

To test the latency of visual stimulus delivery under the graphics processing conditions similar to those of a high-resolution game, this project introduces a 3D Runner game created in UE4, which is a modification of the built-in UE4 template project for first-person gameplay. Gameplay from the 3D Runner can be found on Vimeo [here](file:///C:\Users\iljac\Downloads\something.com). In the game, the player’s first-person character is placed in an arena with several blocks of different sizes. During gameplay, the user is able to control the character to run around the arena freely. At the top right corner of the screen, where the photodiode is attached on the laptop, is a square that changes color from black to white for 11 frames (the stimulus length) and at a preset rate (defining an interstimulus interval, or ISI). This stimulus cycle is activated by left clicking the mouse in-game and appears as a periodic flash of white until the user quits the game. The length of the stimulus was defined in frames due to the high temporal resolution of using UE4’s built-in ‘Tick’ function, which is run once per frame. [25] The stimulus length of 11 frames was chosen due to the fact that it is a factor of the laptop’s refresh rate (165Hz), a design choice adopted from Wiesing, Fink, & Weidner (2020), who also investigated stimulus timing in Unreal Engine. [25] The length was also heuristically chosen to be fast enough to be non-distracting during gameplay, while long enough to generate a robust photodiode response. Because pilot testing for this experiment suggested that a longer ISI may improve delays in stimulus timing, the interstimulus period was also varied between “long” (5s) for one run of testing and “short” (1s) for another. Each run lasted 5 minutes, during which the in-game character and camera were moved around randomly. Finally, the time of each photodiode stimulus onset was timestamped in a CSV file throughout each test.

Another manipulation made to the game was a setting in UE4 called Frame Rate Smoothing. This is a GPU-display synchronization technique similar to a common feature in graphics technology, Vertical Sync (also known as VSync), in that it sets a tight range for in-game frame rate. This is done to reduce the amount of stuttering – or frame rate variability – as the frame rate of the game may be vary widely, depending on resource usage, from the frame rate of the monitor it is run on. In order to study the effects of this setting on the latency of the visual stimulus delivery, the 3D Runner game was run both with the setting on and off.

## Auditory Stimulus Testing

For auditory stimulus testing, the same 3D Runner game was used, however, the stimulus was an audio tone that played periodically rather than a flashing square. Gameplay from this version of 3D Runner can be found [here](file:///C:\Users\iljac\Downloads\something.com). Each of the two tests run used one of two square wave 261.62 Hz (C4 on a piano keyboard) tones generated in Audacity: a “short” stimulus (300ms) and a “long” stimulus (1000ms), both presented once every 3 seconds. These stimulus lengths were chosen to reflect the average duration of stimuli used during in-unit tests. Just like the visual stimulus testing suite, the game was played for 5 minutes, and timestamps of the onset of each auditory stimulus was written to a CSV file throughout each session, while audio data was collected by the microphone.

## Simon Game (Visual & Auditory Stimuli)

The next game implemented for UE4 is a version of the popular electronic game, Simon. In the game, a 2x2 grid of distinctly colored shapes flash (emitting distinct auditory tones and momentary changes in brightness) in a sequence. The goal of the game is for the user to successfully repeat the sequence by pressing on the shapes, after which the sequence is presented once again and extended by one of the four shapes at random. The game, which is theoretically infinite, ends when the user incorrectly inputs the wrong sequence. Being a memory game by design, the main goal of implementing the Simon game for future studies is to investigate the neural mechanisms behind working memory, specifically for encoding language compared to the other features of the in-game stimuli (i.e., position and color). Therefore, in this version of the game, the settings menu includes a plug-and-play audio importer, where the auditory stimuli associated with each button can be defined by the experimenter before running a session. This affords greater flexibility, allowing the experimenter to test neural responses to tones, phonemes, words, non-words, and other auditory stimuli. Also implemented in the game is the flashing square (from the visual stimulus test) in the top right corner of the screen, which flashes at every shape flash the computer creates when presenting a sequence. Furthermore, the game requires more information to be written to the CSV file (color, position, and sound on top of the timestamps of each stimulus onset), which provides an opportunity to observe the effects of an increased information output to the CSV. Gameplay from Simon can be found on Vimeo [here](file:///C:\Users\iljac\Downloads\something.com).

The stimuli used for each square were four 1s tones – 261.62 Hz (C4), 329.62 Hz (E4), 391.99 Hz (G4), and 493.88 Hz (B4) – all generated in Audacity. Prior to gameplay, and upon each incorrect guess (which would cause the game to restart), the color, audio, and position mappings of each square was randomized. This was done in order to dissociate the effects of three features from one another. In other words, future participants could use one or more of the three features to memorize the sequence, and randomizing the mappings is one way of determining which feature might be more informative based on the neural data. Upon beginning gameplay, a startup sequence played in which each square flashed individually and played its corresponding audio. This design decision was adopted from the original electronic game and is done to familiarize the user to the color-audio-position mappings of each button before playing. Finally, there was a 1s delay between each flash in a sequence to make the gameplay around the same speed as the original game.

## Data Analysis

After data collection from the three testing suites, an analysis pipeline written in MATLAB was utilized to extract the stimulus onsets and offsets from either the photodiode or microphone data from the Intan system. A simple threshold crossing algorithm was used to find these onsets and offsets.

There were several metrics used to characterize the timing of stimulus deliveries in the UE4 environments. First, the deviation from the expected ISI was observed for both CSV and Intan data. This was done by subtracting the observed ISI of either source from the expected ISI. Next, deviation from expected stimulus length by the two sources was also investigated and analyzed in the same fashion as ISI deviation. This was only able to be done with Frame Rate Smoothing on, as the frame rate was too variable without the parameter turned on, thus making an “expected length” a poor metric. Specific to the Simon game, the concurrency of visual and auditory stimuli was analyzed. To do this, the time (relative to Intan system) of detected audio tone onsets was subtracted from the time of the simultaneously detected visual flash onsets detected throughout the game. This provided a way to observe the delay between the photodiode impulse sent to the Intan system (marking in “neural time” when a stimulus was shown) and the sound of the actual audio playback. All code for this project is available via GitHub [here](https://github.com/orbitalhybridization/).

# Results

## Visual Stimulus Testing In general, both the CSV and photodiode data exhibit sporadic jumps in timing with respect to the expected interstimulus period (Figures 2-5). These jumps, however, are neither temporally synchronized nor equal in magnitude between the two streams, thus causing an unpredictable latency between them.

It is possible that the distribution of ISI deviation for the photodiode data can be modulated by turning on the Frame Rate Smoothing parameter. Figures 2 and 3 show the deviation from expected ISI for photodiode data without Frame Rate Smoothing, while Figures 4 and 5 show the same metric with Frame Rate Smoothing turned on. Interestingly, Frame Rate Smoothing does not appear to influence the total amount of latency spikes for a given session. Instead, for the smoothed data, the magnitude of the latency spikes appears larger, with the largest being approximately 15-18ms. This is likely a result of the Frame Rate Smoothing lowering the frame rate to a more stable value that the GPU and monitor may struggle less to keep up with. On the other hand, Frame Rate Smoothing appears to have somewhat of an effect on the standard deviation of the ISI deviation, which is generally reduced when the parameter is turned on. Finally, the plots in Figures 2-4 also show an initial ISI deviation of either roughly 17ms (Fig. 3-4) or 6ms (Fig. 2). Knowing the refresh rate of the monitor (165Hz) and the frame rate to which Frame Rate Smoothing defaults (62Hz), these initial delays appear to be single-frame delays.

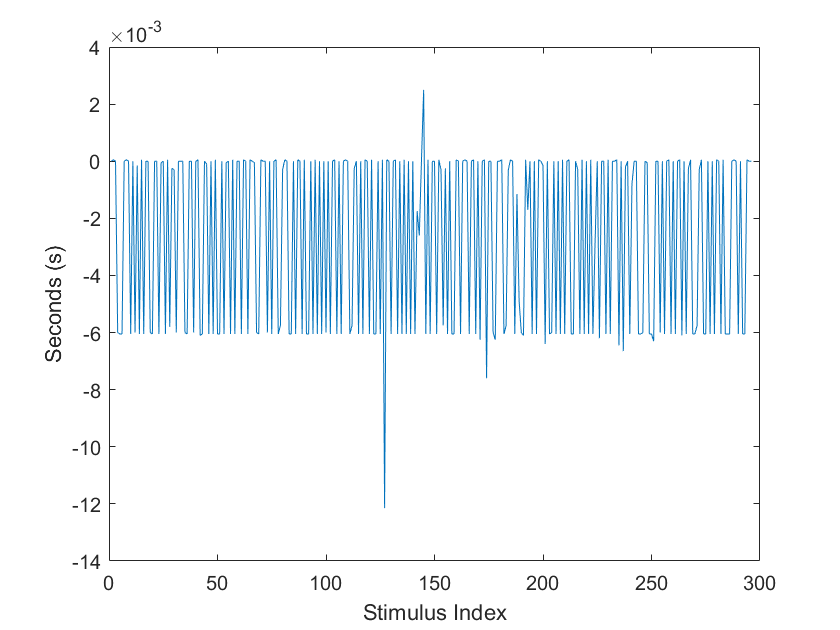


Figure 2. Intan ISI Deviation (ISI=1.0s) – Without Smoothing (**μ=-2.65ms, σ=3.04ms**)

Chart, histogram

Description automatically generated

Figure 3. Intan ISI Deviation (ISI=1.0s) – Without Smoothing (**μ=-1.3ms, σ=2.7ms**)

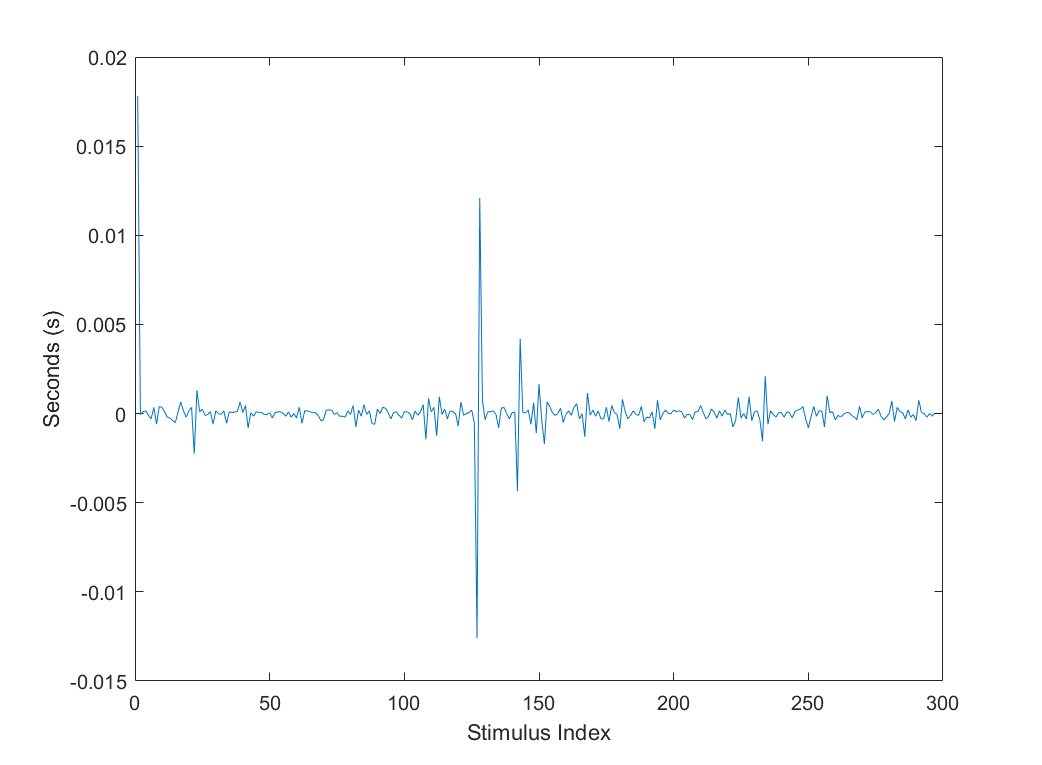


Figure 4. Intan ISI Deviation (ISI=1.0s) – With Smoothing (**μ=0.468ms, σ=1.55ms**)

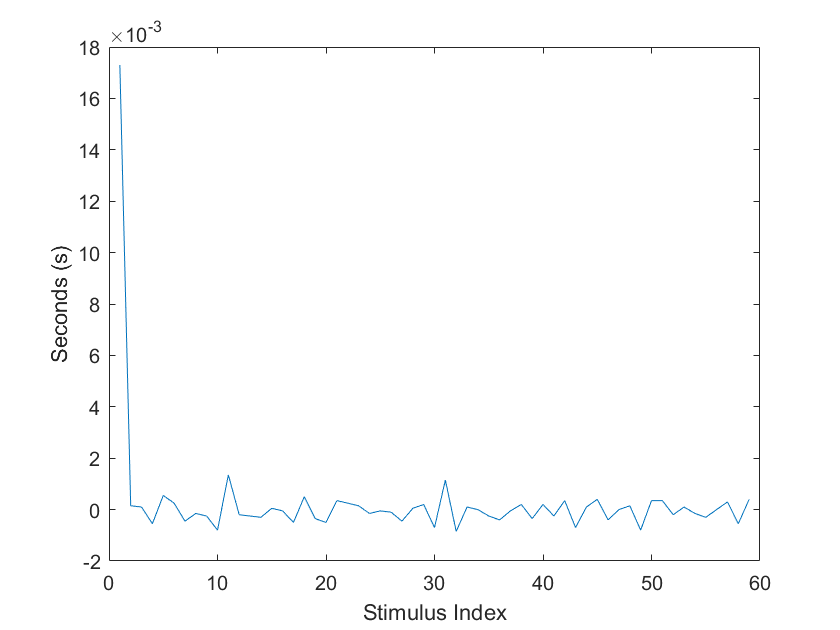


Figure 5. Intan ISI Deviation (ISI=5.0s) – With Smoothing (**μ=0.243ms, σ=2.30ms**)

The CSV logs, on the other hand, are extremely stable relative to the expected ISI, with their standard deviations being in the tenths of milliseconds (Figures 6-7 below). This suggests that the primary cause for latency is the delivery of the photodiode stimulus to the screen, a finding that is paralleled by results from Wiesing, Fink, & Weidner (2020).

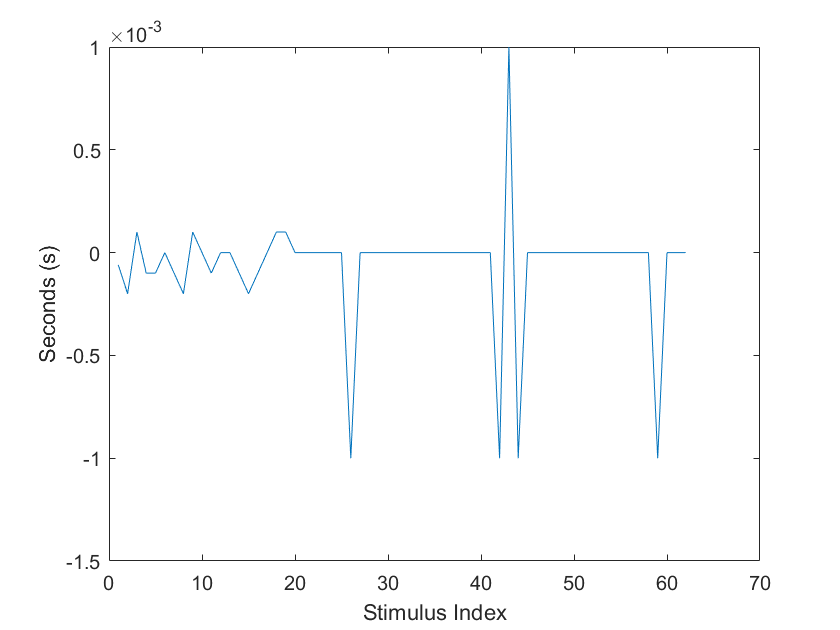


Figure 6. CSV ISI Deviation (ISI=5s) – With Smoothing (**μ=-0.062ms, σ=0.286ms**)

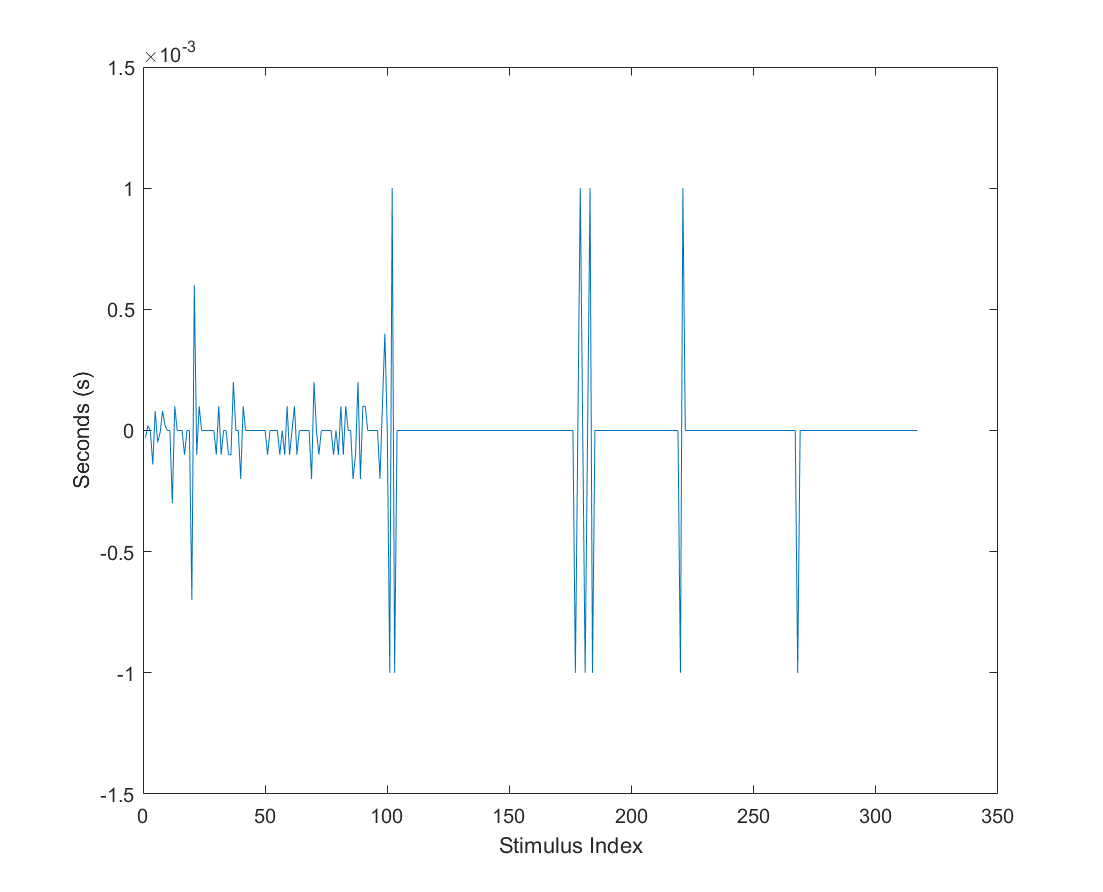


Figure 7. CSV ISI Deviation (ISI=1.0s) – With Smoothing (**μ=-0.012ms, σ=0.200ms**)

Considering that some of the latency spikes are single-frame delays, it is possible that the spikes are caused by a sudden change in the length of the stimulus. Figures 7 and 8 do show that, while most of the detected stimulus lengths are roughly the expected length from the 62Hz smoothed frame rate (where 11 frames equate to 177ms), the length of the stimulus did change sporadically throughout the testing sessions. The indices of these jumps in length also correspond to the indices of spikes in ISI for the ISI of 1.0s test but not for the ISI of 3.0. The spikes, then, are possibly due to a mix of a sudden change in the stimulus length, and small perturbations in the actual frame rate, both of which were discovered in Wiesing, Fink, & Weidner (2020) causing the stimulus delivery to stutter randomly. [25] It is possible, then, that Frame Rate Smoothing reduces the magnitude of that stuttering but does not eliminate it.

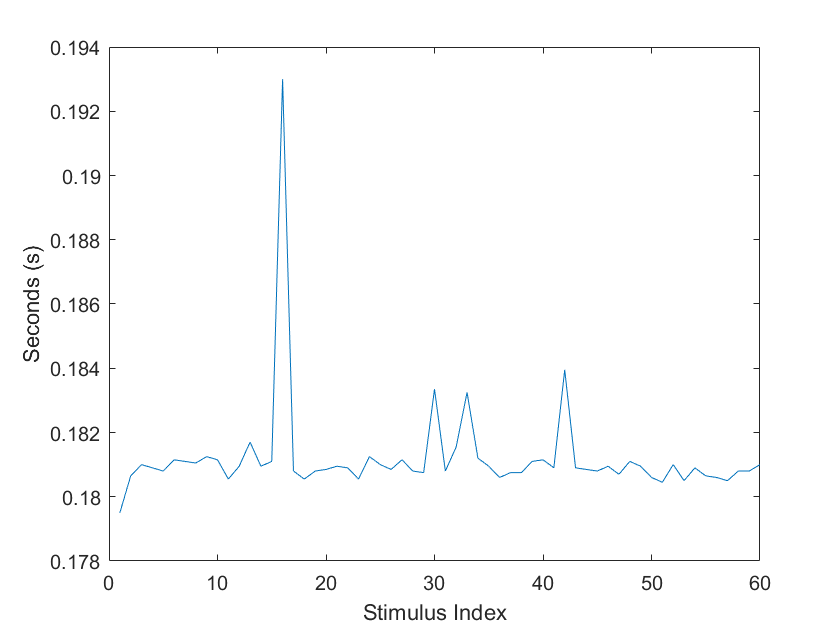


Figure Intan Stim Length Deviation (ISI=5s) (μ=181.2ms, σ=1.7ms)

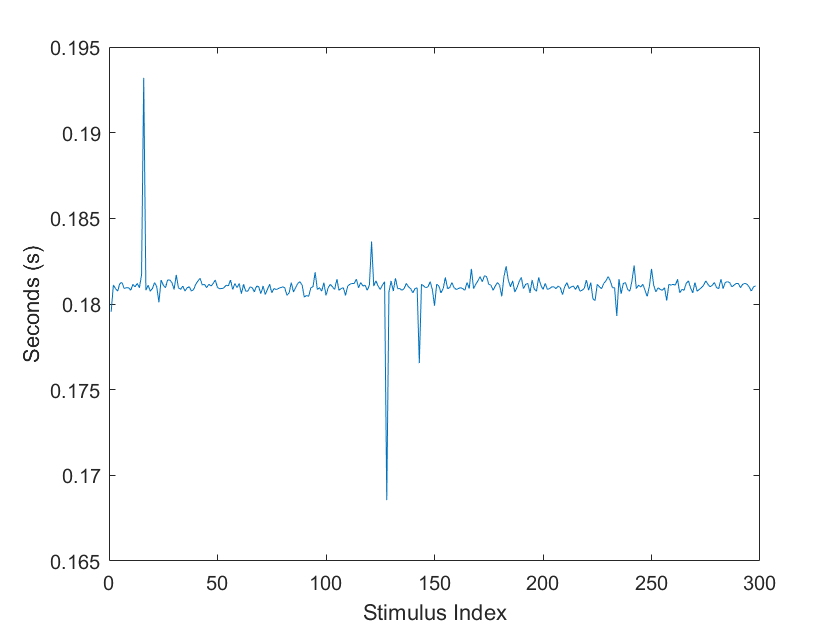


Figure . Intan Stim Length Deviation (ISI=1.0s) (μ=181.0ms, σ=1.1ms)

## Auditory Stimulus Testing

Similar to the visual stimulus testing results, the CSV file logging was found to be more stable (Figures 12-13) than the microphone data picked up by the Intan (Figures 10-11). The auditory stimuli, however, were much more highly variable (σ =11.0ms and 11.3ms) with some of the ISIs deviating as far as 30ms from expected. This change in the ISI distribution could be due to several factors, including the actual sound card component used by the computer. These potential causes are explored in the following section.

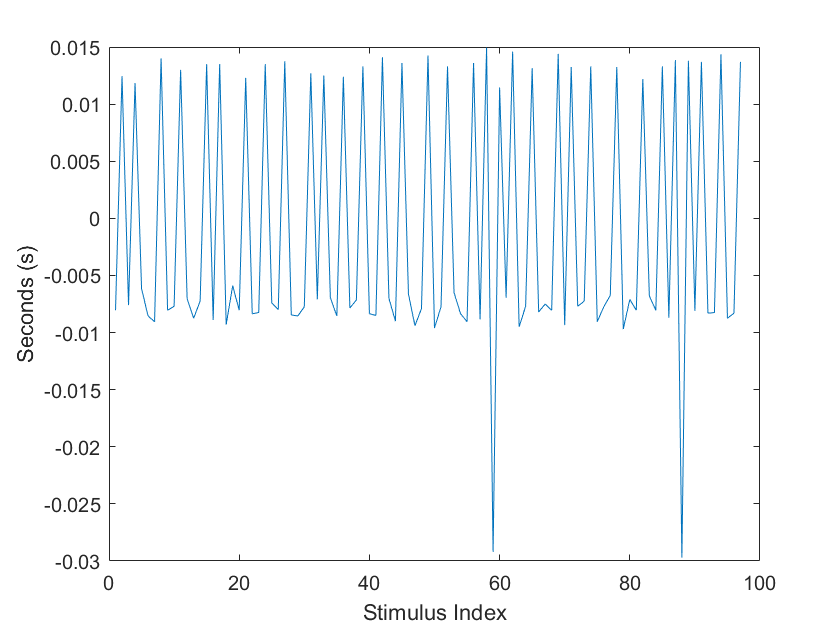


Figure 10. Intan 300ms Tone ISI Deviation (**μ=-1.19ms, σ=11.0ms**)

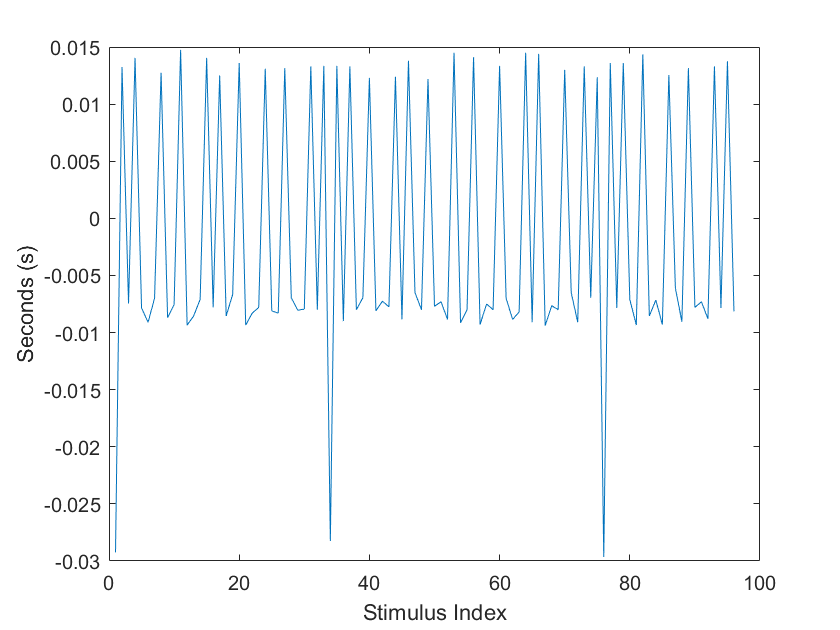


Figure 11. Intan 1000ms Tone ISI Deviation (**μ=-1.5ms, σ=11.3ms**)

Chart, histogram

Description automatically generated

Figure 2. CSV 300ms Tone ISI Deviation (**μ=-1.3ms, σ=0.862ms**)

Chart, histogram

Description automatically generated

Figure 3. CSV 1000ms Tone ISI Deviation (**μ=-1.5ms, σ=0.839ms**)

Looking at how far each detected sound deviated from its expected duration (Figures 14-15), it appears that the longer auditory stimulus was the one that deviated the least. Within these respective distributions, though, there was not much variation in the duration. These results point to a better precision than accuracy when presenting auditory stimuli in UE4, potentially modulated by the stimulus length. A similar result was found in the visual stimulus testing of Wiesing, Fink, & Weidner (2020), where authors discovered a positive relationship between the expected stimulus duration and the error in that presentation.

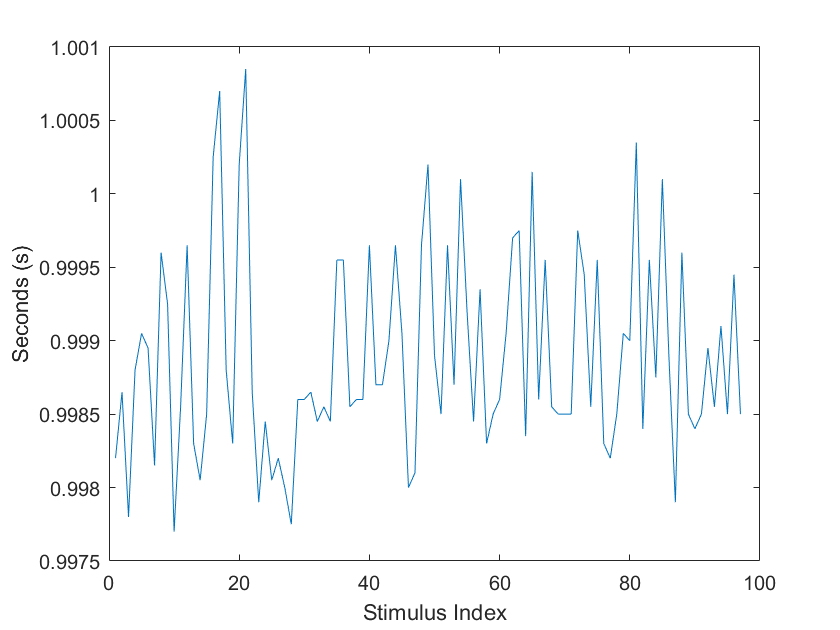


Figure 4. Intan 1000ms Tone Length Deviation (**μ=998.9ms, σ=0.693ms**)

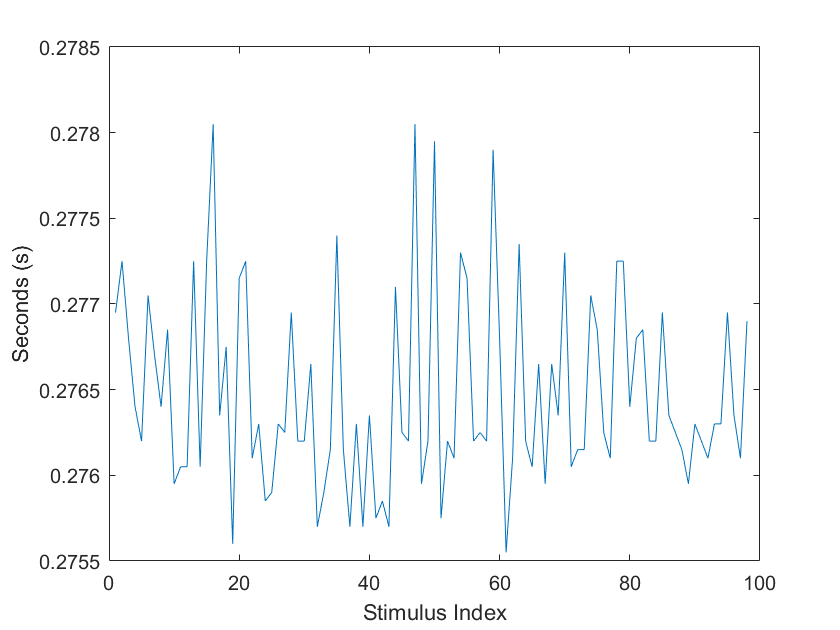


Figure 5. Intan 300ms Length Deviation (**μ=276.5ms, σ=0.566ms**)

## Simon Game

As was the case in the visual stimulus testing, the visual stimulus length varied throughout the Simon game (Figure 16). In this test, however, the visual stimulus length was on average longer than in the previous test, with a mean almost 20 seconds longer than the expected length. This is yet another evidence for the fact that the frame rate, even set by the Frame Rate Smoothing parameter, may not be as expected. Auditory stimulus length (Figure 17), on the other hand, was similar to the previous testing group in both magnitude and stability, confirming earlier results. Most interesting is the latency observed between the microphone and the photodiode data (Figure 18), which can be summarized as an auditory stimulus delay by an average of 125ms *after* the visual stimulus. This latency may be, in part, a biproduct of the previously exemplified internal jitter of auditory presentation (σ ≈11ms), but this cannot account for the entire delay. Other potential sources for this latency are discussed in the following section. Lastly, due to the relative stability of the latency between auditory and visual stimulus detection, it is possible to introduce an offset that would generate an estimate of the actual time of auditory stimulus presentation using only the photodiode.

Chart, histogram

Description automatically generated

Figure 6. Simon Game Visual Stim Length Deviation (**μ=197.6ms, std = 1.1ms**)

Graphical user interface, chart

Description automatically generated

Figure 7. Simon Game Auditory Stim Length Deviation (**μ=998.1ms, std = 1.3ms**)

Graphical user interface, chart

Description automatically generated

Figure 8. Simon Game Visual vs. Audio Stim Timing Difference (**μ=-122ms, std = 7.4ms**)

# Discussion

## Stimulus Timing in Unreal Engine

With a lack of research on stimulus timing tests in Unreal Engine, the primary strength of this project is its addition to the investigations characterizing timing delays in Unreal Engine 4. The results of this project indeed show some promise for the platform to be used in neuroscience research, with some room for further investigation. While a zero latency is ideal, it is an unrealistic expectation given constraints relating to hardware or otherwise. Instead, an error of ±5-8ms is a reasonable margin of acceptance for studying neural correlates. In the visual stimulus domain, the majority of stimuli were shown to be in agreement with these bounds, with the exception of sporadic jumps in stimulus timings. Frame Rate Smoothing seems to be a useful tool for stabilizing the frame rate, but is not completely immune to sporadic delays either, and in fact, may worsen their effect on the data. The story is less promising in the auditory stimulus domain, where a single standard deviation is roughly 11ms, and ISI latency spikes reach up to 30ms in magnitude. The Simon game provided an opportunity to observe audio/visual stimulus concurrency in an asynchronous, naturalistic setting (i.e., free play). Looking at the stimulus concurrency explores the possibility of using the photodiode signal as an anchor for the onset of a stimulus, eliminating the need for a microphone at all. While the two streams did have a 125ms separation on average, this value was relatively stable and could be used to estimate the presentation of a auditory stimulus onset played simultaneously with a visual stimulus. The promise of being able to match the photodiode with the audio data using an offset should be approached with more testing.

So, what could be the cause of these sporadic jumps in stimulus delivery latency? In the visual domain, it was observed that some of the ISI deviation spike indices matched with spikes in stimulus length, similar to a previous study’s finding that actual stimulus length in UE4 varies sporadically (and positively in magnitude) with expected length. [25] This, paired with the fact, that the magnitude of some latency spikes matched the exact length of a frame, suggest that the underlying mechanisms behind these spikes are a combination of both stimulus length and frame rate stuttering. On the audio side of the experiment, the jitter in ISI deviation could be due to several factors, one being that the properties of generated square wave tones are simply more variable when played through UE4. In fact, the more recent version of the software, UE5, uses the native MetaSounds, a higher performance audio system compared to the previous version. UE4, however, was chosen for this study as it is more tested and there exist more development resources are available for that version. The auditory latencies could also be due to internal jitter from the sound card on the stimulation laptop. While the initial results show great promise for UE4’s applications, more investigation must be done in order to further characterize stimulus delivery timings for reliable synchronization with neural data.

## Future Directions

Many of the lab’s in-unit tasks require the press of a button on the laptop to respond to a stimulus. Being able to accurately match the timings of those keypress events on the computer to events in the neural data is also extremely key to investigating neural correlates. This is an advantage that software like Presentation and Psychtoolbox, as the latency of registering keypresses has been tested thoroughly. Extending this kind of testing to Unreal Engine is challenging due to the fact that the software processes keypresses once per frame rather than in parallel to the main thread. This causes uncertain timing delays up to a frame in magnitude, imperceptible to a typical user, but potentially significant in the neural time domain. Authors in Wiesing, Fink, & Weidner (2020) were able to utilize a built-in tool for SteamVR to predict when a stimulus would show on the screen in UE4 and used this to develop a workaround of this limitation. Future iterations of this study could investigate such a workaround using keypress handling tools specific to the Windows OS, likely incorporating Windows API into custom UE4 scripts.

Of course, the eventual goal of this research is to build games that are even more engaging (although Simon may be a classic, it is not nearly as popular as it once was) that also incorporate cognitive skills testing for neuroscience research. Combining the exciting, exploratory aspect of games like Minecraft (Mojang Studios) or Fortnite (Epic Games) with precise, accurate, and research-relevant in-game stimuli would increase patient engagement which, in turn, would improve in-unit study results. This is the huge advantage that Unreal Engine has over standard cognitive neuroscience task presentation software. Though there is room for further investigation, this project was among the first to explore auditory stimulus accuracy and precision in Unreal Engine, and one of few to benchmark stimulus delivery on the platform for use in studying the brain. While just an initial exploration into the potential applications of UE4 for neuroscientific research, the possibilities of those applications are myriad, fueled by artistic vision, engineering practices, and design principles to study the brain.

# References

1. Mooneyham, B. W., Schooler, J. W. (2013). The costs and benefits of mind-wandering: A review. Canadian Journal of Experimental Psychology / Revue canadienne de psychologie expérimentale, 67(1), 11–18. [https://doi.org/10.1037/a0031569](https://psycnet.apa.org/doi/10.1037/a0031569)

2. Manly, T. Robertson I.H., Galloway, M., Hawkins, K. (1998). The absent mind: further investigations of sustained attention to response. *Neuropsychologia, 37*(6), 661-670. <https://doi.org/10.1016/S0028-3932(98)00127-4>

3. Robertson, I.H., Manly, T., Andrade J., Baddeley, B.T., Yiend, J. (1996). ‘Oops!’: Performance correlates of everyday attentional failures in traumatic brain injured and normal subjects. *Neuropsychologia, 35*(6), 747-758. <https://doi.org/10.1016/S0028-3932(97)00015-8>

4. Lelis-Torres, N., Ugrinowitsch, H., Apolinário-Souza, T., Benda, R.N., Lage, G.M. (2017). Task engagement and mental workload involved in variation and repetition of a motor skill. *Scientific Reports, 7,* 14764. <https://doi.org/10.1038/s41598-017-15343-3>

5. Hawkins, G.E., Mittner, M., Forstmann, B.U., Heathcote, A. (2019). Modeling distracted performance. *Cognitive Psychology, 112*, 48-80. <https://doi.org/10.1016/j.cogpsych.2019.05.002>

6. Jongman, S.R., Roelofs, A., Meyer, A.S. (2015). Sustained attention in language production: An individual differences investigation. *Quarterly Journal of Experimental Psychology, 68*(4), 710-730. [https://doi.org/10.1080/17470218.2014.96473](https://doi.org/10.1080/17470218.2014.964736)

7. Ylinen, A., Wikman, P., Leminen, M., Alho, K. (2022). Task-dependent cortical activations during selective attention to audiovisual speech. *Brain Research, 1775*, 147739. <https://doi.org/10.1016/j.brainres.2021.147739>

8. Marsh, J.E., Hughes, R.W., Sörqvist, P. (2015). Dynamic Cognitive Control of Irrelevant Sound: Increased Task Engagement Attenuates Semantic Auditory Distraction. *Journal of Experimental Psychology: Human Perception & Performance, 41*(5), 1462-74. <https://doi.org/10.1037/xhp0000060>

9. Ballesteros, S., Mayas, J., Prieto, A., Ruiz-Marquez, E., Toril, P., Reales, J.M. (2017). Effects of Video Game Training on Measures of Selective Attention and Working Memory in Older Adults: Results from a Randomized Controlled Trial. *Frontiers in Aging Neuroscience, 9.* <https://doi.org/10.3389/fnagi.2017.00354>

10. Chesham, A., Gerber, S.M., Schütz, N., Saner, H., Gutbrod, K., Müri, R.M., Nef, T., Urwyler, P. (2019). Search and Match Task: Development of a Taskified Match-3 Puzzle Game to Asses and Practice Visual Search. *JMIR Serious Games, 7*(2), e13620. https://doi.org/ 10.2196/13620

11. Anderson, J.R., Bothell, D., Fincham, J.M., Anderson, A.R., Poole, B., Qin, Y. (2011). *Journal of Cognitive Neuroscience, 23*(12), 3983-3997. <https://doi.org/10.1162/jocn_a_00033>

12. Stark, C.E.L., Clemenson, G.D., Aluru, U., Hatamian, N., Stark, S.M. (2021). Playing *Minecraft* Improves Hippocampal-Associated Memory for Details in Middle Aged Adults. *Frontiers in Sports and Active Living, 3, 685286.* <https://doi.org/10.3389/fspor.2021.685286>

13. Harrington, C.M., Chaitanya, V., Dicker, P., Traynor, O., Kavanagh, D.O. (2018). Playing to your skills: a randomised controlled trial evaluating a dedicated video game for minimally invasive surgery. *Surgical Endoscopy, 32*, 3813-21. <https://doi.org/10.1007/s00464-018-6107-2>

14. Kral, T.R.A., Stodola, D.E., Birn, R.M, *et al.* (2018). Neural correlates of video game empathy training in adolescents: a randomized trial. *npj Science Learn,* *3*(13). <https://doi.org/10.1038/s41539-018-0029-6>

15. Lohse, K., Shirzad, N., Verster, A., Hodges, A., Van der Loos, H.F.M. (2013). Using Design Principles to Enhance Engagement in Physical Therapy. *Journal of Neurologic Physical Therapy, 37*(4), 166-175. <https://doi.org/10.1097/NPT.0000000000000017>

16. Rupp, M.A., Sweetman, R., Sosa, A.E., Smither, J.A., McConnell, D.S. (2017). Searching for Affective and Cognitive Restoration: Examining the Restorative Effects of Casual Video Game Play. *Human Factors, 59*(7), 1096-1107. <https://doi.org/10.1177/0018720817715360>

17. Welke, D., Vessel, E.A. (2022). Naturalistic viewing conditions can increase task engagement and aesthetic preference but have only minimal impact on EEG quality. *NeuroImage, 256,* 119218. <https://doi.org/10.1016/j.neuroimage.2022.119218>

18. Lin, J., Zhu, Y., Kubricht, J., Zhu, S., Lu, H. (2017). Visuomotor Adaptation and Sensory Recalibration in Reversed Hand Movement Task. *39th Annual Meeting of the Cognitive Science Society.*

19. Laak, K., Vasser, M., Uibopuu, O., Aru, J. (2017). Attention is withdrawn from the area of the visual field where the own hand is currently moving. *Neuroscience of Consciousness,* 2017(1), niw025. <https://doi.org/10.1093/nc/niw025>

20. El-Wajeh, Y.A.M., Hatton, P.V., Lee, N.J. (2022). Unreal Engine 5 and immersive surgical training: translating advances in gaming technology into extended-reality surgical simulation training programmes. *British Journal of Surgery, 109*(5), 470-471. <https://doi.org/10.1093/bjs/znac015>

21. Du, W., Zhong, X., Jia, Y., Jiang, R., Yang, H., Ye, Z., Zong, Z. (2022). A Novel Scenario-Based, Mixed-Reality Platform for Training Nontechnical Skills of Battlefield First Aid: Prospective Interventional Study. *JMIR Serious Games, 10*(4), e40727. https://doi.org/10.2196/40727

22. Garaizar, P., Vadillo, M. A., López-de-Ipiña, D., Matute, H. (2014). Measuring software timing errors in the presentation of visual stimuli in cognitive neuroscience experiments. *PloS one*, *9*(1), e85108. https://doi.org/10.1371/journal.pone.0085108

23. Xie, S., Yang, Y., Yang, Z., He, J. (2005). Millisecond-accurate synchronization of visual stimulus displays for cognitive research. Behavior Research Methods, 37(2). 373-8. https://doi.org/10.3758/bf03192706

24. McKinnery, C.J., MacCormac, E.R., Welsh-Bohmer, K. (1999). Hardware and software for tachistoscopy: How to make accurate measurements on any PC utilizing the Microsoft Windows operating system. https://doi.org/10.3758/BF03207703

25. Wiesing, M., Fink, G., Weidner, R. (2020). Accuracy and precision of stimulus timing and reaction times with Unreal Engine and SteamVR. *PLoS One, 15.* https://doi.org/10.1371/journal.pone.0231152

26. Doucet, G., Gulli, R. A., Martinez-Trujillo, J. C. (2016). Cross-species 3D virtual reality toolbox for visual and cognitive experiments. *Journal of neuroscience methods*, *266*, 84–93. <https://doi.org/10.1016/j.jneumeth.2016.03.009>

27. Robertson, I.H., O’Connell, R. (2010). Vigilant Attention. *Attention and Time,* 79-88. <https://doi.org/10.1093/acprof:oso/9780199563456.003.0006>

28. Brockmyer, J.H. , Fox, C.M., Curtiss, K.A., McBroom, E., Burkhart, K.M., Pidruzny, J.N. (2009). The Development of the Game Engagement Questionnaire: A Measurement of Engagement in Video Game Playing. *Journal of Experimental Social Psychology,* 624-634. <https://doi.org/10.1016/j.jesp.2009.02.016>

1. Sustained attention is defined as “the capacity to maintain accurate responding over time across tasks which can be effortful and demanding, or monotonous and undemanding.” [Robertson & O’Connell, 2010] [↑](#footnote-ref-2)
2. In short, the SART is a go/no-go response task, in which the no-go stimulus is rarely presented. Though it is commonly used as a metric of sustained attention, it is important to note that there is some debate as to whether the SART measures sustained attention as opposed to other cognitive factors such as executive function. [↑](#footnote-ref-3)